REMARKS

Claims 13, 21, and 27 are canceled; thus, claims 1-12, 14-20, and 22-26 are all the claims pending in the application. Claims 1-26 stand rejected on prior art grounds. Applicants respectfully traverse these rejections based on the following discussion.

I.   The Prior Art Rejections

Claims 1-7 stand rejected under 35 U.S.C. §102(e) as being unpatentable over Wiencko, Jr., et al. (U.S. Patent No. 6,557,123), hereinafter referred to as Wiencko. Claims 11-16, 18-24, and 26 stand rejected under 35 U.S.C. §102(b) as being unpatentable over Dunn, et al. (U.S. Patent No. 6,112,255), hereinafter referred to as Dunn. Claim 8 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Wiencko in view of Minow, et al. (U.S. Patent No. 6,021,462), hereinafter referred to as Minow. Claims 9-10, 17, and 25 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Wiencko in view of Sasamoto, et al. (U.S. Patent No. 6,442,711), hereinafter referred to as Sasamoto. Applicants respectfully traverse these rejections based on the following discussion.

The claimed invention provides a method for reliably storing and updating data blocks on a disk array. In the rejection, the Office Action argues that the prior art of record discloses many features of the claimed invention. However, Wiencko does not teach determining which "codes" (which the Office Action asserts teaches the "computation" of the claimed invention) have been updated recently, and updating the "codes" that have not been updated recently prior to updating the "codes" that have been

updated recently. Instead, Wiencko only discloses that the "codes" may be selected (i.e., updated) "each time data is stored on the disk array". In addition, Dunn does not teach determining which data blocks in the failed disk array have been updated recently, and updating the data blocks that have not been updated recently prior to updating the data blocks that have been updated recently. Instead, Dunn merely discloses updating a failed disk drive using parity data from counterpart disk drives of a disk array. Therefore, as explained in greater detail below, Applicants respectfully submit that the prior art of record does not teach or suggest the claimed invention.

A.   The Rejections Based on Wiencko

Applicants traverse the rejections because Wiencko fails to teach or suggest the claimed features of "identifying data blocks of said data blocks that have not been updated recently relative to other data blocks of said data blocks that have been updated recently; [and] on a disk failure in said disk array, updating said computation … and updating said data blocks, wherein said updating of said data blocks comprises updating said data blocks that have not been updated recently prior to updating said data blocks that have been updated recently" as defined in independent claim 1.

The Office Action asserts that the "codes" of Wiencko teach the "computation" of the claimed invention (Office Action, p. 6, item 1). Moreover, the Office Action argues that Wiencko discloses updating the "codes" (Office Action, p. 6, item 1 (citing col. 23, lines 36-50 of Wiencko)). However, Applicants submit that Wiencko does not teach determining which "codes" have been updated recently, and updating the "codes" that

have not been updated recently prior to updating the "codes" that have been updated recently. Instead, Wiencko only discloses that the "codes" may be selected (i.e., updated) "each time data is stored on the disk array" (Wiencko, col. 23, lines 36-50). Nevertheless, nothing within Wiencko mentions determining which "codes" have been updated recently and which "codes" have not been updated recently. Further, nothing within Wiencko mentions giving preferential treatment to the "codes" based on the time since their last update.

As described in column 23, lines 36-50 of Wiencko, referring to FIG. 16.2, a flowchart describing a method for use in redundancy adjustability is shown. Beginning at a start block 1600, control data that is indicative of an amount of data redundancy desired in the disk array is received (step 1602). Based on this control data, one of multiple sets of data redundancy functions (or codes) are selected (step 1604). One of multiple sets of data recovery functions (codes) is selected based on the control data (step 1606). This method may be executed each time data is stored on the disk array (FIG. 17) and each time data is read from the disk array (FIG. 18). However, Wiencko does not discuss the order in which the "codes" are selected.

To the contrary, as described in paragraph 0026 of the claimed invention, an enhancement provided by the system 300 is that it maintains some update statistics (e.g., time since last update) that can be used to predict how likely a block 312 will be updated again in the near future. Such statistics are stored in temporary storage such as semiconductor memory in storage unit 310. When the redundant information stored in blocks 312 is to be updated, the system 300 can use such information to first focus on the

blocks 312 that are less likely to be updated again. For example, because data usage tends to exhibit temporal locality of reference, a block that has been updated recently will tend to be updated again soon. System 300 would therefore first focus on blocks 312 containing redundant that have not been updated recently.

Accordingly, Applicants submit that Wiencko does not teach determining which "codes" (which the Office Action asserts teaches the "computation" of the claimed invention) have been updated recently, and updating the "codes" that have not been updated recently prior to updating the "codes" that have been updated recently. Instead, Wiencko only discloses that the "codes" may be selected (i.e., updated) "each time data is stored on the disk array". However, Wiencko does not disclose the order in which the "codes" are selected.

Therefore, it is Applicants' position that Wiencko fails to teach or suggest the claimed features of "identifying data blocks of said data blocks that have not been updated recently relative to other data blocks of said data blocks that have been updated recently; [and] on a disk failure in said disk array, updating said computation … and updating said data blocks, wherein said updating of said data blocks comprises updating said data blocks that have not been updated recently prior to updating said data blocks that have been updated recently" as defined in independent claim 1. Further, it is Applicants' position that dependent claims 2-7 are similarly patentable, not only because of their dependency from a patentable independent claims, but also because of the additional features of the invention they defined. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections.

B.   The Rejections Based on Dunn

Applicants traverse the rejections because Dunn fails to teach the claimed features of "identifying data blocks of said data blocks that have not been updated recently relative to other data blocks of said data blocks that have been updated recently; and on a disk failure in said disk array, updating said data blocks, wherein said updating of said data blocks comprises updating said data blocks that have not been updated recently prior to updating said data blocks that have been updated recently" as defined in independent claims 11 and 19.

The Office Action asserts that Dunn teaches updating data blocks in a failed disk drive (Office Action, p. 8, item 1(citing Dunn, col. 1, lines 30-41 and 59-65)). However, Dunn does not teach determining which data blocks have been updated recently, and updating the data blocks that have not been updated recently prior to updating the data blocks that have been updated recently. Instead, Dunn merely discloses updating a failed disk drive using parity data from counterpart disk drives of a disk array. However, Dunn does not disclose the order in which data blocks within the failed disk drive are updated.

More specifically, as described in column 1, lines 30-41 of Dunn, Ouchi disclosed segmenting a binary data string into N data blocks, generating a redundancy N+1 st code block (GF(2)) over the string, and writing the data and redundancy blocks out to counterpart N+1 failure-independent disk drives or DASDs. In the event of single disk drive failure, the segment of data stored on the failed drive could be recovered by accessing the remaining N-1 data disk drives and the drive storing the parity segment

image and taking their modulo 2 sum. By use of a simple block parity image, Ouchi could maintain a low redundancy to data DASD ratio. Nevertheless, nothing within Dunn discloses determining which data blocks in the disk array have been updated recently, and updating the data blocks that have not been updated recently prior to updating the data blocks that have been updated recently.

To the contrary, as described in paragraph 0026 of the claimed invention, an enhancement provided by the system 300 is that it maintains some update statistics (e.g., time since last update) that can be used to predict how likely a block 312 will be updated again in the near future. Such statistics are stored in temporary storage such as semiconductor memory in storage unit 310. When the redundant information stored in blocks 312 is to be updated, the system 300 can use such information to first focus on the blocks 312 that are less likely to be updated again. For example, because data usage tends to exhibit temporal locality of reference, a block that has been updated recently will tend to be updated again soon. System 300 would therefore first focus on blocks 312 containing redundant that have not been updated recently.

Accordingly, Applicants submit that Dunn does not teach determining which data blocks in the failed disk array have been updated recently, and updating the data blocks that have not been updated recently prior to updating the data blocks that have been updated recently. Instead, Dunn merely discloses updating a failed disk drive using parity data from counterpart disk drives of a disk array. However, Dunn does not disclose the order in which data blocks within the failed disk drive are updated.

Therefore, it is Applicants' position that Dunn fails to teach the claimed features of "identifying data blocks of said data blocks that have not been updated recently relative to other data blocks of said data blocks that have been updated recently; and on a disk failure in said disk array, updating said data blocks, wherein said updating of said data blocks comprises updating said data blocks that have not been updated recently prior to updating said data blocks that have been updated recently" as defined in claims independent claims 1 and 19. Further, it is Applicants' position that dependent claims 12-16, 18, 20-24, and 26 are similarly patentable, not only because of their dependency from a patentable independent claims, but also because of the additional features of the invention they defined. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections.

C.   The Rejections Based on Wiencko and Minow

As discussed above, Wiencko fails to teach or suggest the claimed features of "identifying data blocks of said data blocks that have not been updated recently relative to other data blocks of said data blocks that have been updated recently; [and] on a disk failure in said disk array, updating said computation … and updating said data blocks, wherein said updating of said data blocks comprises updating said data blocks that have not been updated recently prior to updating said data blocks that have been updated recently" as defined in independent claim 1 (from which claim 8 depends upon).

Further, Applicants submit that Minow is introduced by the Office Action for the mere purpose of illustrating skipping the updating and deleting process if a set of

retrievable addresses exceeds a fraction of the data stored in the disk array (Office Action, pp. 10-11, item 2). Minow discloses updating "main memory" by transferring data from secondary storage devices. However, Minow does not disclose determining which data blocks in the "main memory" have been updated recently, and updating the data blocks that have not been updated recently prior to updating the data blocks that have been updated recently. Instead, Minow merely discloses updating the "main memory" upon receiving a "data transfer request" (Minow, col. 7, lines 54-67). In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections.

      D.   The Rejections Based on Wiencko and Sasamoto

As discussed above, Wiencko fails to teach or suggest the claimed features of "identifying data blocks of said data blocks that have not been updated recently relative to other data blocks of said data blocks that have been updated recently; [and] on a disk failure in said disk array, updating said computation ... and updating said data blocks, wherein said updating of said data blocks comprises updating said data blocks that have not been updated recently prior to updating said data blocks that have been updated recently" as defined in independent claims 1, 11, and 19 (from which claims 9-10, 17, and 25 depend upon, respectively).

Further, Applicants submit that Sasamoto is introduced by the Office Action for the mere purpose of illustrating that altered portions of the computed function are updated whenever a load on the disk array is below a threshold value (Office Action, p.

11, item 2). Moreover, the Office Action references Sasamoto to illustrate a method wherein altered portions of the computed function that are less likely to be altered again are preferentially updated (Office Action, p. 12, item 2). However, Sasamoto does not teach determining which data blocks in the disk array have been updated recently, and updating the data blocks that have not been updated recently prior to updating the data blocks that have been updated recently. Instead, Sasamoto "evenly updates" the data blocks. More specifically, as described in column 6, lines 56-60 of Sasamoto, when the host computer 100 requests to write data during the reproducing the data for the preventive maintenance, the disk array control unit 101 <u>evenly updates</u> both of the disk drive 1 and the disk drive 4 as long as the data is stored in an area which is already reproduced. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections.

II.   Formal Matters and Conclusion

In view of the foregoing, Applicants submit that claims 1-12, 14-20, and 22-26, all the claims presently pending in the application, are patentably distinct from the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary. Please charge any

deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.

Respectfully submitted,

Dated: <u>December 20, 2007</u>

/Duane N. Moore/
Duane N. Moore
Registration No. 53,352

Gibb & Rahman, LLC
2568-A Riva Road, Suite 304
Annapolis, MD 21401
Voice: (410) 573-6501
Fax: (301) 261-8825
Customer Number: 29154